DRUKWERK

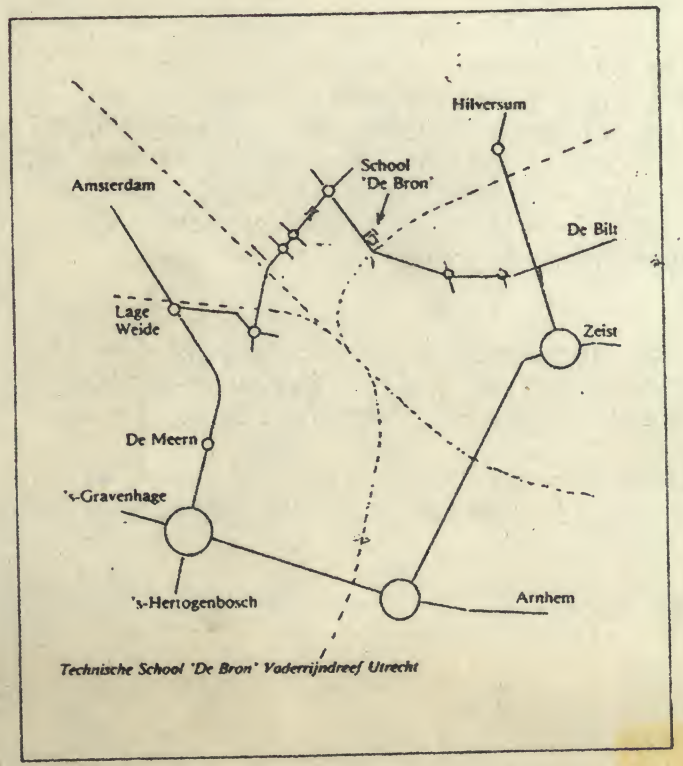
PORT BETAALD
LEIDEN

afzender: 68000 gg, abrikozenstr. 86, 2564VT Den Haag

68KARANT

nummer 8, augustus 1985, verschijnt 5X p.j.

Adres: Technische School 'De Bron' Vaderrijndreef 7 Utrecht (Overvecht)



Technische School 'De Bron' Vaderrijndreef Utrecht

Met openbaar vervoer:
Naar Utrecht CS, met buslijn 7 naar Overvecht, uitstappen halte bij de school (vraag maar aan de chauffeur). Met de trein naar station Overvecht, 5 minuten lopen naar 'De Bron'. Vanuit het station recht oversteken, pad volgen in richting flat links aanhouden, spoor volgen tot aan 'De Bron'.

Met de auto:

Vanuit *Amsterdam*

Afslag 'Lage Weide', richting Centrum op verkeersplein scherp linksaf, onder 2 spoorwegviaducten door, kruising rechtdoor, 2 rotondes rechtdoor, brug over, op verkeersplein rechtsaf, 2e verkeerslicht linksaf, school ligt aan de rechterhand.

Vanuit *'s-Gravenhage*
Op Ouderrijn, richting Amsterdam inslaan, 2e afslag is 'Lage Weide'. Zie verder boven.

Vanuit *'s-Hertogenbosch*
Na 'Ouderrijn', 2e afslag is 't Lage Weide'. Zie verder hierboven.

Vanuit *Arnhem*
Afslag Utrecht Amersfoort, daarna bij afslag Utrecht De Bilt richting Utrecht, 1e rotonde rechtdoor, spoor over, 2e rotonde rechtdoor, bij splitsing rechts aan, onder spoor door, rechtsaf bij verkeerslicht, school ligt aan de rechterhand.

Vanuit *Hilversum*
Afslag Groenekam, Enthovendreef, linksaf Darwindreef, Eijkmanlaan, op Eijkmanlaan rechtsaf onder spoor door, rechtsaf bij verkeerslicht, school ligt aan de rechterhand.

Van de redaktie

Het verschijnen van deze achtste 68KArant zal bijna gelijktijdig zijn met nummer 74 van de HCCnieuwsbrief, met daarin een artikel over ATARI's nieuwe 68000 machine, de 520ST. Door dit artikel dat meer tijd kostte dan ik had verwacht, en tentamenperikelen is deze 68KArant een beetje in het gedrang gekomen.

De planning was een uitgebreide enquette te maken en met deze 68KArant mee te sturen. Het is de bedoeling dat er een adreslijst komt die aangevuld wordt met een aantal gegevens uit de enquette in gekodeerde vorm,zodat mensen elkaar beter weten te vinden. Nu de enquette is uitgesteld is er gelegenheid om suggesties aan te dragen, bv. wat u van andere leden goed lijkt om te weten (op komputergebied). Verder hopen we dan wat meer informatie te krijgen over wat er aan verwachtingen en wensen leeft tav. de 68000 gebruikersgroep en ook om een indruk te krijgen van interessante dingen waar leden mee bezig zijn om ze vervolgens wat aan hun jasje te kunnen trekken om er wat meer ruchtbaarheid aan te geven.

Redaktie 68KArant.
Om de basis van de redaktie wat te verbreden ben ik op zoek gegaan naar informatika studiegenoten hier in Leiden. Rene van Kesteren wil ik hierbij even voorstellen. Hij studeert wiskunde en informatika en woont in Noordwijk. Hij is bezig mijn MC 68000 systeem (uit het duitse blad MC) af te bouwen, waar ik de laatste maanden geen tijd voor had. Hij heeft eerder de Junior komputer uit elektuur gebouwd en is al ruime tijd geinteresseerd in de 68000.
Verder heeft Guus Ramackers mij al geholpen bij het schrijven van het artikel over de 520ST in de HCCNieuwsbrief. Hij studeert ook informatika en woont eveneens in Noordwijk. Inde toekomst wil hij meer met de 68000 doen en aan publikaties bijdragen.

68008 op eurokaart.
Peter Ebbelink heeft een systeempje ontworpen dat op een enkele eurokaart past. In samenwerking met onze sekretaris Robert Smink en Penningmeester Jan Blok is hiervoor een print ontworpen (op een 68000 cad/cam systeem !). Ik heb begrepen dat er op het kaartje naast de 68008 vier 28-pens icvoetjes voor ROM en RAM,een 68681 met oa.twee luxe seriele poorten en een 68230 met parallele i/o en timer zitten. Binnenkort zal er wel wat over gepubliceerd worden.

wandelgangen.
-afgelopen bijeenkomst hoorde ik van een fout in de schakeling die je nodig hebt als je een IBM-toetsenbord aan het MC zelfbouwontwerp wilt knopen. Er was me een korrekt schema toegezegd, wat ik alsnog graag tegemoet zie.
-RTOS het RealTime Operating Systeem van het c't zelfbouwsysteem wil alleen opstarten als er een interuptlijn aangesloten is begreep ik van Willem van Spronsen uit Leiden.

HCC-dagen.
De HCC dagen op op 22 en 23 november zijn nog even weg, maar toch moeten we al eens gaan nadenken wat we daar willen doen. Mij lijkt het een goed idee om er een diakast neer te zetten. Als er nu mensen zijn die een leuke 68000 opstelling hebben, die ze niet mee naar de HCCdagen (kunnen) nemen, dan zou een of een paar dia's daarvan een oplossing kunnen bieden . Er rest nog enige tijd om uw diarolletje vol te maken ! Ik heb al dia' s van de lege MC en C't printen en ben van plan nog een serie dia's van de 520ST te maken. Andere ideen zijn zeer welkom.

Bijeenkomst
Op 31 augustus in de technische school de bron te Utrecht, vanaf 11 uur in een lokaal op de eerste etage. De volgende zaken zullen hoogst waarschijnlijk te zien zijn: ATARI 520ST, kompleet c't bouwpakket van R. v.d.Kamp,MC-systeem en het ontwerp van het 68008 kaartje van P.Ebbelink.

Tom den Duijf,morsstraat 50, 2312BN Leiden, 071-134371.

Groningen,15/07/1985.

NDR-III Klein Computer deel 2.

Na het eerste deel in de 68Karant van maart hier 'n vervolg.
De floppy kaart is al reeds wat langer binnen,kosten DM390,-,en werdt beschreven in MC no.2/1985. (Processor 1797,dataseparator 9229.Men kan vier drives,gemengd 51/4'' en 8'').Na sparen voor een disk drive werdt ook het kleine Operating System in twee 8k-eprom's leverbaar,kosten DM90.Hiermee kan men diskettes formatteren in SD-40tracks met 5 sectoren van elk 1024 bytes (MC standaard), DD-80 tracks idem en SD 77 tracks 8'' met 26 sectoren van elk 256 byte.Dit formatteren gebeurt per zijde,welke men dan ook steeds moet opgeven.
Men kan teksten,data en grafieken opslaan en weer laden op namm,en eventueel vanaf een ander adres laten schrijven.Met het commando DIR 1,0 krijgt men de inhoudsopgave van drive 1,zijde 0.Totaal zijn 170 namen mogelijk.Met de cursor kan men het gewenste programma aanwijzen,en wordt dit na <cr> automatisch geladen op het oude adres.De editor wordt ook naar dit adres gezet.Indien men het programma op een ander adres wil,kan dat mbv.commando's TLOAD,DLOAD en GLOAD waarbij men naast naam,drive en zijde ook startadres moet aangeven.
Wissen van files gebeurt met 'SCRATCH'.Het lezen en schrijven van losse sectoren is ook mogelijk.De 'HELP' functie ontbreekt ook niet.Het OS programma wordt aangeroepen via de bibliotheek functie.Bepaalde subroutineadressen kan men snel in de symbooltabel laten zetten zodat men deze in andere programma's kan oproepen.Zo kan men makkelijk een groter OS bouwen,voortbordurend op het reeds bestaande.Afgezien van het wennen aan de diverse termen werkt het geheel al weer zo'n twee maanden naar behoren.
Als aanvulling hierop is een los formatteeer programma in 8kEPROM leverbaar, (DM75),waarmee men vele gangbare diskette-formaten kan gebruiken.
Als tweede OS is CP/M68K incl.C-compiler,utilities,assembler etc.,leverbaar voor DM775 incl. documentatie.Hiermee heeft men dan aansluiting op een standaard pakket software.Het geheel is (nog?) niet in mijn bezit.
Verder zijn leverbaar:
-seriele poort (DM140).
-256k-RAM kaart(ongeveer DM450)
-16-voudige AD-omzetter met 8 bit resolutie(DM150).
-enkelvoudige AD omzetter met 10 bit resolutie (265).
-dubbele DA omzetter,8 bit (DM140).
-soundgenerator rond de AY-3-8912 (DM90).
-disassembler in 8k EPROM (DM55)

Mischien kom ik op de volgende meeting met de computer,zodat alles te zien is hetgeen natuurlijk veel fraaier is.

gegroet

Eelco van der Wal.

ZEEF VAN ERATOSTHENES OP SAGE-II

Inleiding

In navolging van het bekende test programma uit Byte [1] om priem getallen te berekenen tussen 3 en 8191, heb ik deze test met verschillende tot mijn beschikking staande programmeertalen uitgeprobeerd op een Sage-II [2].

De Sage-II computer is een micro gebaseerd op een 68000 (8 MHz klok) met 512 Kbyte RAM geheugen, 2 dubbelzijdige 5 1/4 inch floppy disk drives, 2 serial poorten, een centronics uitgang en een IEEE-488 interface (vergelijk [3]).
Als operating systemen zijn o.a. verkrijgbaar: UCSD p-System (standaard), Volition's Modula-2 [4], CP/M 68K, Idris etc.

Programmeertalen

Als programmeertaal is UCSD-Pascal toegepast (p-System versie IV.13), welke zowel p(seudo)-code als n(ative)-code kan leveren. Beiden worden door het operating system ge-interpreteerd, waarbij de n-code bijna overeenkomt met machine code en dus aanzienlijk sneller is dan p-code (gemiddeld ca. 10 keer).
Onder Volition's Modula-2 operating system (compiler versie 0.3e) is de programmeertaal Modula-2 (zie ook eerdere 68Karanten) beschikbaar, die p-code genereerd, die op z'n beurt weer wordt ge-interpreteerd. Als laatste is het programma geschreven in 68000 assembly, waarbij gebruik is gemaakt van de 68000 assembler behorende bij het p-System.

Test programma's

De drie programma's in Pascal, Modula-2 en assembly zijn weergegeven in resp. listing 1, 2 en 3. Hierbij moet worden opgemerkt dat het Pascal en Modula-2 programma 10 keer dezelfde priemgetallen berekenen en het assembly programma doet dit 100 keer. Dit is gedaan om van de laatste nauwkeuriger de executie tijd (alles is met de hand geklokt) te kunnen bepalen. (Het aantal gevonden priemgetallen moet 1899 bedragen.)

Het resultaat van de testen is in tabel 1 vermeld. De tijden in de tabel zijn het gemiddelde van steeds 5 runs, terwijl de tijd bij "assembly" het gemiddelde van 5 runs gedeeld door 10 is.
De kolom "range checking" geeft aan of gebruik is gemaakt van de "range check option". Indien deze actief is (default) dan genereerd de compiler extra code om runtime te kunnen testen of er niet buiten de array grenzen wordt geadresseerd.

test	programmeertaal	range checking	10 iterations [s]
1	Pascal p-code	nee	61.0
2	idem	ja	71.0
3	Pascal n-code	nee	5.0
4	idem	ja	5.0
5	Modula-2 p-code	nee	56.0
6	idem	ja	66.0
7	6800 assembly	-	1.9

tabel 1

Volgens de gemeten tijden scheelt dit 10 s (ca. 15%) t.o.v. de p-code versies en niets (niet waarneembaar) t.o.v. de n-code versie. Het verschil in executie tijd tussen p- en n-code (Pascal) is goed te zien (ca. 12 keer sneller).

De assembly versie is slechts 2,5 keer sneller dan de n-code versie. Volgens de testresultaten in BYTE [1] moet echter 0.49 s (!) haalbaar zijn op een 8 MHz 68000 (wie doet een poging?).

Conclusie

Met dit test programma wordt slechts ten dele de rekenkracht van een (micro) computer aangetoond. Veel hangt af van de efficiency van de compiler en de interpreter, maar door vergelijking met testen op andere machines mag gesteld worden dat de Sage tot de snelste 16-bitters behoort.

Referenties

1. Gilbreath, J. en G.Gilbreath,
   Eratosthenes Revisted,
   BYTE, januari 1983, p 283
2. Rietschote, H.F. van,
   Test Sage-II,
   Databus, december 1983, p 40
3. Spronsen, W.A. van,
   Sage-IV,
   68Karant, januari 1985
4. Joyce, E.,
   Volition's Modula-2 on the Sage
   BYTE, september 1984, p 351

W.N.Jacobs

```
Pascal Compiler IV.13 c6t-4      5/25/85

    1   2    1:d    1  program sieve;
    2   2    1:d    1
    3   2    1:d    1  {
    4   2    1:d    1   eratosthenes sieve prime number program
    5   2    1:d    1   ref: byte jan 1983, p 283 e.v.
    6   2    1:d    1   }
    7   2    1:d    1
    8   2    1:d    1   const
    9   2    1:d    1     size = 8190;
   10   2    1:d    1
   11   2    1:d    1   var
   12   2    1:d    1     flags: array [0..size] of boolean;
   13   2    1:d 8192     i,prime,K,count,iter: integer;
   14   2    1:d 8197
   15   2    1:0    0   begin
   16   2    1:1    0     writeln('10 iterations');
   17   2    1:1   20     for iter := 1 to 10 do begin
   18   2    1:3   38       count := 0;
   19   2    1:3   42       for i := 0 to size do
   20   2    1:4   61         flags[i] := true;
   21   2    1:3   84       for i := 0 to size do
   22   2    1:4  103         if flags[i] then begin
   23   2    1:6  118           prime := i + i + 3;
   24   2    1:6  130           {writeln(prime);}
   25   2    1:6  130           K := i + prime;
   26   2    1:6  140           while (K <= size) do begin
   27   2    1:8  149             flags[k] := false;
   28   2    1:8  163             K := K + prime
   29   2    1:7  166           end;
   30   2    1:6  175           count := count + 1
   31   2    1:5  178       end;
   32   2    1:2  191     end;
   33   2    1:1  201     writeln(count,' primes')
   34   2     :0    0  end.

End of Compilation.
```

listing 1

```
    1   7  1:D    0 (*$TO "list2.text";*)
    2   7  1:D    1 MODULE Sieve;
    3   7  1:D    1
    4   7  1:D    1 (*eratosthenes sieve prime number program
    5   7  1:D    1  byte jan 1983, p 283 e.v.*)
    6   7  1:D    1
    7   7  1:D    1 FROM InOut IMPORT WriteLn, WriteString, WriteCard;
    8   7  1:D    1
    9   7  1:D    1 CONST
   10   7  1:D    1   SIZE = 8190;
   11   7  1:D    1
   12   7  1:D    1 VAR
   13   7  1:D    1   flags: ARRAY [0..SIZE] OF BOOLEAN;
   14   7  1:D8192   i,prime,K,count,iter: CARDINAL;
   15   7  1:D8197
   16   7  1:C    0  BEGIN
   17   7  2:C    0    WriteLn; WriteString('10 iterations');
   18   7  2:C   30    FOR iter := 1 TO 10 DO
```

```
21  7  2:C  42          flags[i] := TRUE
22  7  2:C  54        END;
23  7  2:C  66        FOR i := 0 TO SIZE DO
24  7  2:C  70          IF flags[i] THEN
25  7  2:C  85            prime := i * 2 + 3;
26  7  2:C  95            (*WriteCard(prime, 5);*)
27  7  2:C  95            k := i + prime;
28  7  2:C 105            WHILE (k <= SIZE) DO
29  7  2:C 115              flags[k] := FALSE;
30  7  2:C 129              INC(k,prime)
31  7  2:C 137            END;
32  7  2:C 139            INC(count)
33  7  2:C 144          END;
34  7  2:C 144        END;
35  7  2:C 154      END;
36  7  2:C 162      WriteLn; WriteCard(count,6); WriteString(' primes')
37  7  1:C 189 END Sieve.
```

listing 2

```
0000:                                   .PROC   SIEVE
0000:                                 ;*****************************************************************
0000:                                 ; Eratosthenes Sieve Prime Number Program in 68000 Assembler
0000:                                 ; (SAGE-II with 8 Mhz clock, no wait states)
0000:                                 ;*****************************************************************
0000:                                         .INCLUDE        PUS1:SAGE.ASM.TEXT
0000:                                 ;*********************************************************
0000:                                 ; Common part to interface with Sage prom functions and
0000:                                 ; Sage BIOS
0000:                                 ;*********************************************************
0000:                                 ;
0000:                                 ; long jump to subroutine to SAGE prom entry
0000:                                 ; (FE0000H + offset)
0000:                                 ;
0000:                                         .MACRO  LJSR
0000:                                         .WORD   4EB9H
0000:                                         .WORD   00FEH
0000:                                         .WORD    %1
0000:                                         .ENDM
0000:                                 ;
0000:                                 ; long jump to SAGE prom entry
0000:                                 ; (FE0000H + offset)
0000:                                 ;
0000:                                         .MACRO  LJMP
0000:                                         .WORD   4EF9H
0000:                                         .WORD   00FEH
0000:                                         .WORD    %1
0000:                                         .ENDM
0000:                                 ;
0000:                                 ; Sage prom entries
0000:                                 ;
0000: 0008                            KEYBCH  .EQU     8H     ;get a keyboard character
0000: 000C                            KEYCHK  .EQU    0CH     ;check for a keyboard character
0000: 0014                            TERMCHAR.EQU    14H     ;output a character to the terminal
0000: 0018                            TERMTEXT.EQU    18H     ;output a text string
0000: 001C                            TERMCRLF.EQU    1CH     ;print a carriage return/line feed
0000: 0020                            TERMHEXB.EQU    20H     ;output a hexadecimal byte
0000: 0024                            TERMHEXW.EQU    24H     ;output a dexadecimal word
0000: 0028                            FDREAD  .EQU    28H     ;floppy disk read
0000: 002C                            FDWRITE .EQU    2CH     ;floppy disk write
0000: 0030                            BOOTSX  .EQU    30H     ;floppy disk boot
0000: 0038                            DEBUG   .EQU    38H     ;debugger entry point
0000:
0000:                                  ;
0000: 7003                                    MOVEQ   #3.,D0                  ;print start messages
0002: 4E4E                                    TRAP    #14.
0004: 41FA ****                               LEA     STARTMES,A0
0008:                                         LJSR    TERMTEXT
0008: 4EB9                            #       .WORD   4EB9H
000A: 00FE                            #       .WORD   00FEH
000C: 0018                            #       .WORD    TERMTEXT
000E:                                         LJSR    TERMCRLF
000E: 4EB9                            #       .WORD   4EB9H
0010: 00FE                            #       .WORD   00FEH
```

```
0812: 081C                      #       .WORD   TERMCRLF
0814:                           ;=======================================================
0814:                           ; Target independent part
0814:                           ;
0814:                           ; register assignment:
0814:                           ;
0814:                           ;       D0 : iteration counter (iter)
0814:                           ;       D1 : prime counter (count)
0814:                           ;       D2 : temporary counter (i,j)
0814:                           ;       D3 : prime number (prime)
0814:                           ;       D4 : loop counter (k)
0814:                           ;       A0 : base flag array (flag)
0814:                           ;
0814: 1FFE                      SIZE    .EQU    8190
0814: 0000                      TRUE    .EQU    0
0814: 0001                      FALSE   .EQU    1
0814:
0814: 7063                              MOVEQ   #100-1,D0       ;setup iteration count (iter)
0816: 41FA ****                         LEA     FLAGS,A0
081A: 4241                      PRIM6   CLR.W   D1              ;prime count = 0
081C:                           ;
081C: 343C 1FFE                         MOVE.W  #SIZE,D2        ;setup loop counter (i)
0820: 4230 2000                 PRIM1   CLR.B   TRUE(A0,D2)     ;flag[i]=true
0824: 51CA FFFA                         DBF     D2,PRIM1        ;only decrement
0828:                           ;
0828: 4242                              CLR.W   D2              ;j=0
082A: 4A30 2000                 PRIM5   TST.B   TRUE(A0,D2)
082E: 6600 ****                         BNE     PRIM2
0832: 3602                              MOVE.W  D2,D3           ;flag[j]=true
0834: D642                              ADD.W   D2,D3
0836: 5643                              ADDQ.W  #3,D3           ;prime=j+j+3
0838: 3803                              MOVE.W  D3,D4
083A: D842                              ADD.W   D2,D4           ;k=j+prime
083C: B87C 1FFE                 PRIM4   CMP.W   #SIZE,D4
0840: 6200 ****                         BHI     PRIM3
0844: 11BC 0001 4000                    MOVE.B  #FALSE,0(A0,D4) ;flags[k]=false
084A: D843                              ADD.W   D3,D4           ;k=k+prime
084C: 60EE                              BRA     PRIM4
084E:                           ;
0842* 000C
084E: 5241                      PRIM3   ADDQ.W  #1,D1           ;count=count+1
0830* 0020
0850: 5242                      PRIM2   ADDQ.W  #1,D2           ;j=j+1
0852: B47C 1FFE                         CMP.W   #SIZE,D2
0856: 66D2                              BNE     PRIM5
0858: 51C8 FFC8                         DBF     D0,PRIM6
085C:                           ;
085C:                                   LJSR    TERM%CRLF       ;print #primes
085C: 4EB9                      #       .WORD   4EB9H
085E: 00FE                      #       .WORD   00FEH
0860: 081C                      #       .WORD    TERMCRLF
0862: 3001                              MOVE.W  D1,D0
0864:                                   LJSR    TERM%HEXW
0864: 4EB9                      #       .WORD   4EB9H
0866: 00FE                      #       .WORD   00FEH
```

```
0068: 0024                      0       .WORD   TERM!EXW
006A: 41FA ****                         LEA     ENDMES,A0
006E:                                   LJSR    TERMTEXT
006E: 4EB9                      0       .WORD   4EB9H
0070: 00FE                      0       .WORD   00FEH
0072: 0018                      0       .WORD   TERMTEXT
0074:                                   LJSR    TERMCRLF
0074: 4EB9                      0       .WORD   4EB9H
0076: 00FE                      0       .WORD   00FEH
0078: 001C                      0       .WORD   TERMCRLF
007A:                           ;=======================================================
007A:
007A: 4240                              CLR.W   D0
007C: 4E4E                              TRAP    #14.            ;entry SAGE debugger
007E:
0018* 0066
007E: 00 00 00 00 00 00 00 00   FLAGS   .BLOCK  SIZE+1,TRUE     ;byte array (flags)
2078:
2078: 31 30 30 20 69 74 65 72   STARTMES.ASCII  "100 iterations"
2085: 61 74 69 6F 6E 73
2088: 00                                .BYTE   0
208C: 20 70 72 69 6D 65 73      ENDMES  .ASCII " primes"
2093: 00                                .BYTE   0
2094:
2094:                                   .END
```

listing 3

Overzicht dokumentatie 68000 artikelen, en aanverwanten.
Datum laatste mutatie : mei, 11 1985

Indeling : \
Artikel koplijn,
Bron van oorsprong,
gekregen van,
Kort overzicht.

=====
001 FPLA forms VME Bus interrupter
EDN, dec 1983

Opbouw van interrupt logika met een FPLA
=====
002 32 bit prozessor ersetzt 8 bit CPU
Elektronik aug. 1983

Beschreven wordt hoe een 68008, met wat randlogika, op de plaats van een 6809 gezet kan worden.
=====
003 The VW68K Single-board-computer
BYTE jan. 1984

Beschrijving van VU68K SBC kit van $200 met VUBUG monitor.
Met hardware, en software overzicht.
Inklusief source van de VUBUG monitor
=====
004 Verkuerzter WAIT-zyclus beschleunigt 68000 systeme
Elektronik feb. 1984

(L. v. Breda)
Optimaliseren van wait m.b.v. een kleine ekstra schakeling
=====
005 Powerful VME Bus features ease high-level uC applcations
EDN jan. 1984

(L. v. Breda)
Inleiding in VME bus spec's, met overzicht van beschikbare kaarten van diverse fabrikanten, met een lijst kontakt adressen m.b.t. VME
=====
006 DMA controller takes aim at 16-bit systems
Electronic Design okt. 1983

(L. v. Breda)
Memory management concept en DMA uitgelegd.
=====
007 SAGE-II mini of micro ?
Databus dec. 1983

Beschrijving 68K systeem van f16.000,=
=====
008 An MC68000 overvieuw
MICRO - The 6502/6809 Journal sep. 1982

(T. Nicola)
De 68000 uitgelegd. Alleen deel 1; algemeen
=====
009 68000 Instruction set
MICRO - The 6502/6809 Journal sep. 1982

(T. Nicola)

Overzicht van de 68K instrukties met een kleine uitleg
bij verschillende instrukties

=====

010 The 68000 and the personal computer
MICRO - The 6502/6809 Journal sep 1982

(T. Nicola)
Enige argumentering om je 6502 te verbouwen tot 68000

=====

011 8-bit-Prozessor bietet 32-bit-architektur
Elektronik jan. 1983

(T. Nicola)
Grof overzicht/inleiding 68008

=====

012 Ein 16-bit Mikroprozessor in HMOS technik
Elektronik ??

(T. Nicola)
Grof overzicht technologie, opbouw, instruktie set,
performance, software koncept

=====

013 Echtzeit-emulation fuer die M68000 familie
Elektronik sep 1983

(T. Nicola)
Kort overzicht EXORmacs en HDS-400 emulator

=====

014 CPU der 68000-familie unterstuetzt virtuellen Speicher
Elektronik nov. 1983

(T. Nicola)
Kennismaking 68010

=====

015 32-bit prozessor erweitert 68000 familie
Elektronik mrt 1983

(T. Nicola)
Kennismaking 68020

=====

016 Einchip computer ist zur 68000 familie kompatibel
Elektronik jul 1983

(T. Nicola)
Kennismaking 68200 (CPU, ROM, RAM, I/O)

=====

017 'Fremde' interface baustein an der CPU68000
Elektronik aug. 1983

(T. Nicola)
Ideen om AmZ8530, AmZ8538, AmZ9513, Am9516, Am9519 (resp.
seriele komm. contr., FIFO, programmeerbare interval timer,
DMA controller en programmeerbare interrupt contr.)
aan de 68000 te knopen.

=====

018 VME-bus kompatibles aufbausystem
Elektronik mrt. 1983

(T. Nicola)
Samenvatting normering VME mechanisch opbouw

=====

019 Universeller 68000 VME uC bietet minicomputer leistung
Elektronik jun. 1983

(T. Nicola)

Single board computer met 68000 van MicroSys. De kaart is op dubbele eurokaart met VME interface, en bus. Tevens is een lokale bus beschikbaar.
Monitor: micromon.

=====

020 Flexibel, vielseitig und leistungsfaehig
Elektronik okt. 1983

(T. Nicola)
Introduktie VME-10 tafelkomputer

=====

021 68K monitor (source)
Veel kommentaar. Uit een boek

=====

022 68K monitor van KWS computer systeme (source)

Duits talig.

=====

023 VUBUG 68K monitor (source)

Van Am. zelfbouw projekt. Stond ook in BYTE.
Inklusief handleiding.

=====

024 MC68020.32-bit Prozessor fuer zukunftsichere systemkonzepte
Elektronik juli 1985

Inleiding 68020 en vergelijking met 68000/68010

=====

025 Aktualisierte uebersicht der VME-baugruppen
Elektronik jan. 1984

Overzicht VME kaarten met firma adres lijst

=====

026 VMEbus-Computer: universell und flexibel
Elektronik mei, 1984

Bespreking enkele VME kaarten

=====

027 UNIX fuer die M68000-familie
Elektronik juni 1985

Kort verhaal over UNIX met 68K processor

=====

028 The VU68K single-board computer
BYTE jan 1984

Beschrijving 68K single board comp. voor zelfbouw

=====

029 Using the 68008
wireless world nov 1983

(RJ v/d Kamp)
Vervangen van een 6809 door een 68008

=====

030 Designing with the 68008 microprocessor
wireless world apr. 1984

(RJ v/d Kamp)
Basis informatie onwikkelen van hardware voor 68008

=====

031 An EPROM simulator
BYTE maart 1985

(RJ v/d Kamp)

SRAM met bus buffers en battery backup

=====

032 A low cost, low write-voltage EEPROM
BYTE feb 1984

(RJ v/d Kamp)
Hoe de 52B13 (intel:2817) te gebruiken

=====

033 Multibus II: 32 bit bus fuer leistungsfaehige offene systeme
Elektronik jan 1984

Korte inleiding multibus

=====

034 PROM programmiergeraet fuer VME bus
Elektronik apr. 1984

Hardware beschrijving programmeerapparaat voor diverse populaire PROMs

=====

B00 MC68120/68121 intelligent peripheral controller user's manual
B01 Single chip microcomputer data 84/85
B02 motorola 16/32 bit mcu system components
B03 motorola microprocessor software catalog
B04 CMOS data manual standard logic 1
B05 CMOS dat manual special functions
B06 MCU/MPU applications manual vol 2
B07 M68000 16/32 bit microprocessor programmers reference manual 4th edition
B08 Telecommunications data manual

=====

data sheets:
MC68008
MC68440
MC68451
MC68681
MC68901

Procedure

Dokumentatie aanvraag telefonisch (010-100386)
dan neem ik het de volgende bijeenkomst mee.
Het kan aldaar gekopieerd worden of meegenomen worden.
Ik verwacht het de ledenbijeenkomst daarna weer terug in mijn bezit.
Per post gaat altijd voor rekening van de aanvrager(heen en terug).

Peter Ebbelink.

De test op de SAGE is afkomstig van: W.N.Jacobs
Gasthuislaan 28
6883JD Velp